

ELECTRONIC MAIL SOFTWARE WITH MODULAR INTEGRATED AUTHORING/READING
SOFTWARE COMPONENTS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to an electronic mail program. More particularly, the invention relates to an electronic mail program having modular integral authoring/reading applications whereby documents created with the modular integral authoring/reading applications are seamlessly sent and received by the mail program.

2. State of the Art

In recent years electronic mail ("email") has become widely used in business, education, and in personal communications. One of the features of electronic mail which is most convenient, particularly in business and in education, is the ability to attach a binary computer file to an email message. This feature enables email correspondents to rapidly share word processing documents, database documents, spreadsheet documents, multimedia documents, or virtually any kind of binary file created by a computer. There are, however, some serious limitations and inconveniences associated with attaching a binary file to an email message.

1 The original Internet mail system as defined in 1982 with RFC
2 (Request for Comments) 821 and 822 had a number of important
3 limitations. In particular, the system was not designed to carry
4 large quantities of arbitrary data in an email message. In fact,
5 the 1982 SMTP (Simple Mail Transport Protocol) standard required
6 that an email message consist of a single message containing only
7 ASCII characters in lines of 1000 characters (blocks of 32k) or
8 less. Moreover, some implementations of SMTP or other mail
9 transport systems (such as UUCP) restricted message lengths to
10 some allowed maximum number of bytes. Messages passing through a
11 mail gateway using one of these implementations were likely to be
12 truncated.

13
14 The ability to send large quantities of binary data through
15 the Internet electronic mail system was made possible with the
16 MIME (Multipurpose Internet Mail Extensions) standard for
17 Internet messages. The original MIME standard was published as an
18 Internet Request For Comments document (RFC 1341) and approved in
19 June of 1992. (See Internet RFCs 2045, 2046, and 2047 for the
20 latest MIME standards documents.) The MIME standard describes
21 how an email message should be formatted in order to be considered
22 MIME compliant. MIME defines a set of message header fields and a
23 set of message encoding standards that are designed to overcome
24 the limitations of RFC 822 message formats and still be

1 transportable through any of the numerous legacy mail transport
2 systems in use on the Internet. MIME message header fields extend
3 those defined in RFC 822 and describe the content and encoding
4 type of the email message. Encoding schemes allowed in the MIME
5 standard include "quoted-printable", and "base64". In addition,
6 three unencoded data types are allowed. These are labeled "8bit",
7 "7bit", or "binary".

8
9 If the sender and the receiver of the email message with the
10 attached binary file are using the same brand and version of email
11 program and both programs are configured in substantially the same
12 way, the receiver's email program should automatically apply the
13 appropriate decoding to the attached binary file and produce a
14 file which is identical to the file which was attached to the
15 email by the sender. However, if the sender and receiver are
16 using different email programs, the recipient may receive a file
17 which must be decoded by the recipient using a separate decoding
18 program. Worse yet, if there is a failure of the receiving email
19 program to properly deal with the MIME protocol, it is possible
20 that the receiver will receive multiple files (each being $\leq 32k$)
21 which must first be concatenated and then decoded.

22
23 Even after the file is properly received and decoded, it is
24 often difficult for the receiver of the file to open the file.

1 The receiver of the file might expect that "clicking" on the file
2 icon will open the file. However, clicking on the file icon will
3 often not open the file. It may result in an error message like
4 "application not found" or, worse, it may result in the file being
5 opened by an inappropriate application thereby displaying
6 "gibberish". The receiver of the file must have a program capable
7 of reading (opening) the file. For example, if one attaches a
8 spreadsheet file to an email message, the receiver of the file
9 must have a spreadsheet program in order to open the file.
10 Technically, it is not necessary that the receiver of the file
11 have the same brand program as that which created the file.
12 However, opening a file with a program which did not create it,
13 though possible, can be very inconvenient. The receiver of the
14 file must know what kind of file is attached to the email message,
15 must know what program on their computer is capable of reading
16 that type of file, must launch the program, must open the file
17 from within the program, and wait while the program translates the
18 file.

19
20 The limitations of Internet electronic mail can become even
21 more frustrating if the sender and recipient are not using the
22 same operating system (OS). Some mail attachment encoding schemes
23 (and file compression schemes) are OS-dependent and it is possible

1 that an email recipient could receive a file which is impossible
2 to decode (or decompress).

3
4 These limitations in electronic mail have discouraged many
5 people, particularly non-sophisticated computer users, from
6 attaching files to electronic mail messages. In fact, for some
7 novice users, the task of launching one application to create a
8 document, saving the document, launching a separate email
9 application to create an email message, and then locating the
10 saved document for attachment to an email message is daunting
11 enough to discourage them. In addition, novice users often
12 complain that after "downloading" a file attached to an email
13 message they cannot find the file on their hard disk.

14 15 SUMMARY OF THE INVENTION 16

17 It is therefore an object of the invention to provide an
18 electronic mail program which includes integrated authoring
19 software whereby a document may be created and sent by email in a
20 seamless manner.

21
22 It is also an object of the invention to provide an
23 electronic mail program which includes integrated

1 authoring/reading software whereby a document may be received and
2 opened in a seamless manner.

3
4 It is another object of the invention to provide an
5 electronic mail program which includes modular integrated
6 authoring software whereby different kinds of documents may be
7 created and sent by email in a seamless manner.

8
9 It is still another object of the invention to provide an
10 electronic mail program which includes modular integrated
11 authoring/reading software whereby different kinds of documents
12 may be received and opened in a seamless manner.

13
14 It is another object of the invention to provide an
15 electronic mail program which includes modular integrated
16 authoring/reading software whereby the authoring/reading software
17 and the email software present an interface which suggests that a
18 single application is operating.

19
20 It is another object of the invention to provide an
21 electronic mail program which includes modular integrated mailbox
22 handling software whereby messages of different types are
23 displayed in different ways in a mailbox listing.

1 It is still another object of the invention to provide an
2 electronic mail program which includes modular integrated
3 authoring/reading software wherein the functionality of the
4 authoring/reading software is controlled by the "role" of the user
5 when participating in an exchange of messages.
6

7 In accord with these objects which will be discussed in
8 detail below, the electronic mail software of the present
9 invention includes a main email component and a number of
10 installable components which communicate bidirectionally with the
11 email component. The installable components include
12 authoring/reading components as well as at least one mailbox
13 browser/editor component. The main email component provides an
14 underlying graphical user interface (GUI) for functions directly
15 associated with the storage and transfer of electronic mail
16 messages. In particular, the main email component provides menu
17 items which allow the user to SEND, READ, REPLY, FORWARD, DELETE,
18 SAVE, PRINT, for example. The main email program also handles all
19 data bundling and unbundling that may be required to transform a
20 message created by an authoring component into a fully MIME
21 compliant message. In addition, the main email component includes
22 "hooks" (an application programming interface or API) for the
23 attachment of the installable components. The authoring/reading
24 components each provide functionality which is particular to the

1 type of document the component is designed to create/display. For
2 example, a text document authoring component includes word
3 processing functionality such as font selection, margin setting,
4 etc. A painting/drawing authoring component includes tools for
5 line drawing, polygon creation, paint brush, paint can, eraser,
6 etc. A spreadsheet authoring component displays a grid and
7 includes formula creation tools as well as formatting tools. A
8 database authoring tool includes tools for creating fields and
9 records, for sorting and searching, for generating reports, etc.
10 A photo editor authoring component includes various imaging
11 editing tools including cropping tools, dodging and burning tools,
12 filters, etc. A presentation authoring component includes tools
13 for creating slides and slide shows. The authoring components act
14 like applications embedded within the email program and allow
15 specific types of documents such as spreadsheets, graphics,
16 databases, etc. to be created from within the email program and
17 emailed directly. In addition, the authoring components allow
18 received spreadsheets, graphics, databases, etc. to be read by the
19 email program without the difficulties traditionally associated
20 with attaching binary files to an email letter. According to the
21 invention, in lieu of authoring components which allow both
22 authoring and reading, separate components may be provided for
23 authoring and reading, or components for reading only may be
24 provided in addition to components which permit authoring as well

1 as reading. The authoring/reading components interface with the
2 main email component via designated "MIME types". The MIME data
3 standard allows developers to define MIME types using the label
4 "/application-x" in the data header. The authoring components of
5 the invention pass data to the main email component which packages
6 the data as a MIME compliant message with the label "/application-
7 x" in the message header, where x identifies the authoring/reading
8 component which created/can display the message. When the message
9 is received, the main email component concatenates and decodes the
10 MIME message, reads the MIME type, sends the data to the component
11 associated with the MIME type, and waits for a user event or a
12 callback from the component. This bidirectional communication
13 between the main email component and the authoring/reading
14 components provides a totally seamless operation wherein the user
15 may send and receive complex documents without any knowledge of
16 attaching files, downloading, decoding, etc.

17
18 The mailbox browser/editor (mailbox handler) component is
19 provided preferably as a separate component rather than as part of
20 the main email component so that the software may be more easily
21 customized and upgraded. The mailbox browser/editor component is
22 used to display, edit, and browse mailboxes. Since the invention
23 provides for email messages which contain different kinds of data,
24 the features of the mailbox browser may depend on the type of

1 messages being sent and received. For example, if a graphical
2 authoring/reading component is installed, it may be desirable to
3 provide a mailbox browser which shows a thumbnail of the selected
4 graphic email message when a list of messages is displayed.

5
6 The software according to the invention provides a single
7 seamless environment for authoring, reading, and emailing a
8 variety of different types of documents. The user does not need
9 to understand uploading, downloading, file types, file decoding,
10 or any of the other esoteric requirements of attaching files to
11 email. Further, the user does not need to know what kind of
12 application must be launched in order to read a particular type of
13 email message.

14
15 An exemplary embodiment of the invention is an email program
16 for school children called KIDCODE®. The KIDCODE® program
17 includes a main email component, a mailbox browser/editor
18 component and several message authoring/reading components. The
19 main email component and the mailbox browser/editor component
20 provide the same functionality as described above. Additional
21 KIDCODE® components include a text authoring tool, rebus game
22 message handler components (encoding and decoding components)
23 which allow children to create and respond to graphical rebus
24 messages, several different game puzzle components, and a workbook

1 which allows a teacher to send workbook problems to a student and
2 allows the student to send the solved problems back to the
3 teacher. According to one inventive aspect of the invention which
4 is exemplified in the workbook and rebus components, an
5 authoring/reading component may assign and track user "roles" by
6 associating a role tag to each message. For example, in the rebus
7 component, the user initiating the rebus exchange will be assigned
8 the role of rebus encoder. The message created by this user will
9 contain a tag identifying it as an "encoded message". When the
10 message is opened by the recipient, the tools available in the
11 rebus component will be different from those available if a
12 message were being encoded. Similarly, the workbook component is
13 preferably provided with a teacher role and a student role, each
14 of which have different tools. Component roles may be selected by
15 the users, assigned by the system administrator, or automatically
16 by components when messages are created/read.

17
18 According to a presently preferred embodiment, the KIDCODE®
19 client software is written in the MACROMEDIA DIRECTOR™ LINGO™
20 scripting language which is cross-platform and thus ideally suited
21 for use in schools which typically have a combination of MAC/OS™
22 and WINDOWS™ computers. As implemented, the client software
23 operates over a TCP/IP LAN which is the most common type of
24 network used in schools today and is compatible with the Internet.

1 According to a further implementation of the invention, KIDCODE®
2 software permits messages to be sent via the Internet in MIME
3 compliant format.

4
5 Additional objects and advantages of the invention will
6 become apparent to those skilled in the art upon reference to the
7 detailed description taken in conjunction with the provided
8 figures.

9
10 BRIEF DESCRIPTION OF THE DRAWINGS

11
12 Figure 1 is a screen shot of the KIDCODE® client login
13 screen;

14
15 Figure 1a is a simplified state diagram illustrating the
16 entry from the login screen to the main email component of the
17 KIDCODE® software;

18
19 Figure 2 is a screen shot of the KIDCODE® main email
20 component screen showing a menu of the installed authoring/display
21 components and the mailbox browser component;

22
23 Figure 2a is a simplified state diagram illustrating the
24 entry from the main email component into the installed components;

1 Figure 3 is a screen shot of the KIDCODE® mailbox
2 browser/editor component screen;

3
4 Figure 4 is a screen shot of the KIDCODE® text message
5 authoring component screen;

6
7 Figure 5 is a screen shot of the KIDCODE® rebus authoring
8 (encoding) component screen;

9
10 Figure 6 is a screen shot similar to Figure 5 illustrating a
11 listbox of users on the network to whom mail may be sent;

12
13 Figure 7 is a screen shot similar to Figure 5 illustrating a
14 rebus in the process of being coded by the user;

15
16 Figure 8 is a screen shot of the KIDCODE® rebus reading
17 (decoding) component screen;

18
19 Figure 9 is a screen shot of the KIDCODE® workbook authoring
20 component screen;

21
22 Figure 10 is a screen shot illustrating the main email
23 component of a second embodiment of the invention;

24

1 Figure 11 is a screen shot illustrating a text authoring
2 component in the second embodiment of the invention;

3
4 Figure 12 is a screen shot illustrating a painting/drawing
5 authoring component in the second embodiment of the invention;

6
7 Figure 13 is a screen shot illustrating a spreadsheet
8 authoring component in the second embodiment of the invention;

9
10 Figure 14 is a screen shot illustrating a database authoring
11 component in the second embodiment of the invention;

12
13 Figure 15 is a screen shot illustrating a photo editor
14 authoring component in the second embodiment of the invention;

15
16 Figure 16 is a screen shot illustrating a slide show
17 authoring component in the second embodiment of the invention; and

18
19 Figure 17 is a screen shot illustrating a display-only
20 component in the second embodiment of the invention.

1 BRIEF DESCRIPTION OF THE APPENDICES

2
3 Appendix A is the LINGO™ script implementation of the
4 KIDCODE® main email component;

5
6 Appendix B is the LINGO™ script implementation of the
7 KIDCODE® mailbox handler component;

8
9 Appendix C is the LINGO™ script implementation of the
10 KIDCODE® text authoring/displaying component;

11
12 Appendix D is the LINGO™ script implementation of the
13 KIDCODE® rebus game coding/decoding component;

14
15 Appendix E is a description of the Application Programming
16 Interface for the KIDCODE® main email component which enables
17 installable components to operate with the main email component;
18 and

19
20 Appendix F is a description of and pseudocode for the
21 Internet implementation of the KIDCODE® software.
22
23

1 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

2
3 As mentioned above, a presently implemented embodiment of the
4 invention is realized utilizing the MACROMEDIA DIRECTOR LINGO™
5 scripting language. The DIRECTOR™ application was originally
6 intended to be used for authoring interactive multimedia
7 presentations. Therefore, many of the terms used in the LINGO
8 language refer to multimedia objects such as movies, cast members,
9 frames, and sprites. However, it has been found that the LINGO™
10 language can be used to author many different types of programs
11 including programs which were not traditionally thought of as
12 being multimedia presentation programs. The following
13 description, therefore, of the presently implemented embodiment
14 will be best understood by those familiar with the MACROMEDIA
15 DIRECTOR LINGO™ scripting language. However, those skilled in the
16 art will understand from the functional description which follows
17 that the invention could be implemented in other languages such as
18 C or C++, JAVA™, etc.

19
20 Referring now to Figures 1 and 1a, and with reference to
21 Appendix A, the first screen 10 presented by the KIDCODE® program
22 is preferably a login screen where the user enters his or her name
23 and password. According to the presently preferred embodiment,
24 the login name field 12 is implemented as a popup menu (or pull

1 down list box) and the password field 14 is a standard text entry
2 field. See, for example, lines 172-190 of Appendix A. Clicking
3 on the login name field will make a list of names appear and allow
4 the user to highlight one of the names using the mouse. After the
5 user has selected a name and typed in a password, the Okay button
6 16 must be clicked, or the Return or Enter key may be pressed.
7 See Appendix A, lines 796-846 and lines 879-899. At this screen
8 10, the only option available under the FILE menu is QUIT.
9 According to the presently preferred embodiment, the usernames and
10 passwords are stored in associative (two property) lists so that a
11 password may be associated with a username and a username may be
12 associated with a password. When the okay button is clicked, the
13 software checks the validity of the username and password. The
14 checking of the username and password is illustrated in the state
15 diagram in Figure 1a. Starting at 20 in Figure 1a, if the user
16 selects a username and clicks the okay button, the password field
17 is checked at 22. If no password was entered, a popup message is
18 displayed at 24 indicating to the user that a password must be
19 entered and the system returns to start at 20 waiting for the user
20 to click the okay button. If the user types in a password and
21 clicks the okay button, the username field is checked at 26. If
22 no username was selected, a popup message is displayed at 24
23 indicating that a username must be selected and the system returns
24 to start at 20 waiting for the user to click the okay button. If

1 the user types in a password and selects a username, it is
2 determined at 28 or 30 whether the username and password match,
3 and if they do, the software enters to the main email component at
4 32. If the username and password do not match, a popup message is
5 displayed at 34 indicating that the password entered is invalid
6 for the username selected and the system returns to start at 20
7 waiting for the user to click the okay button. If the username
8 and password are that of the system administrator, a special
9 administration display will be shown in which usernames and
10 passwords may be added/deleted to/from the system. See Appendix A
11 lines 858-875, 900-1016, and 1123-1140.

12
13 Turning now to Figures 2 and 2a, once the user has selected a
14 username and entered the correct password, the program displays
15 the screen 40 shown in Figure 2 (Appendix A lines 851-855). This
16 is the screen of the main email component with no other component
17 selected. The screen 40 includes a scrollable collection of icons
18 42, 44, 46, 48, 50 and includes buttons 52, 54, 56 for mailbox
19 access and button 58 to quit the program. The icons 42, 44, 46,
20 48, and 50 represent the installed authoring/reading components.
21 As shown in Figure 2, the icons represent a text component 42, a
22 rebus component 44, a "text in grid" component 46, a puzzle
23 component 48, and a "connect the dots" component 50. As
24 illustrated in Figure 2a, starting at 60, if the user clicks on

1 the text icon, the program will, at 62, open the text authoring
2 component with an empty message (Appendix A lines 1422-1438 and
3 1025-1054). Similarly, if the user clicks on the rebus icon, the
4 program will, at 64, open the rebus authoring component with an
5 empty message (Appendix A lines 1442-1458 and 1025-1054). The
6 main email component will also, when an authoring component is
7 opened, make the print, trash, and send buttons visible as these
8 functions are served by the main email component as illustrated in
9 Figure 2a (Appendix A lines 489-526 and 1456). Since the
10 presently implemented example does not yet have all components
11 complete, the launching of other components is shown generically
12 at 66 in Figure 2a. When a component is launched, the main email
13 program suspends execution at 68 awaiting any "call back" from the
14 launched component. The API described in Appendix E includes a
15 set of "call back" functions that can be used by an authoring or
16 mailbox component to communicate with the main email component.
17 See Appendix E pages 4-7 and Appendix A lines 557-744. Further,
18 the main email component serves the email functions of inbox,
19 outbox, and filing cabinet for messages that have not been sent.
20 These functions are illustrated in Figure 2 as buttons 52, 54, 56
21 and in Figure 2a as routines 72, 74, 76.

22
23 For example, as shown in Figure 3, when the user enters the
24 inbox, a window 78 and a button bar 80 are displayed. The mailbox

1 component with the appropriate set of messages is launched as
2 shown in Appendix A lines 1533-1558 and 216-247). The window 78
3 displays a list of new email and the button bar 80 displays
4 buttons for functions common to all components, i.e. buttons for
5 reply 82, send 84, print 86, and trash 88. It will be understood
6 that depending on whether the user is in an authoring or reading
7 mode, either the reply button 82 or the send button 84 will be
8 "grayed out" indicating that that option is not available. As
9 shown in Figure 3, for example, the send button 84 is not
10 available when looking at the mail inbox. It will also be
11 appreciated that the buttons and icons from the previous screen
12 (Figure 2) are no longer visible. As stated in Figure 2a, the
13 user returns to the screen of Figure 2 when the window 78 is
14 closed. Appendix B lines 440-448 and Appendix A lines 735-744.

15
16 The presently implemented text authoring/reading component is
17 illustrated in the authoring mode in Figure 4 and is described in
18 detail in Appendix C. The window 90 is similar to any email
19 message authoring tool and includes fields for "to:" 92, "from:"
20 94, "date" 96, "subject" 98, and "message" 100. The "from:" field
21 94 and "date" field 96 are hidden in Figure 4 behind the
22 scrollable list box 102. After the addressee is chosen from the
23 list box 102, the box disappears and reveals the "from" and "date"
24 fields. According to the presently preferred embodiment, the list

1 box 102 lists the names of all of the users registered in the
2 local email system. This is handled by a call to the API as
3 indicated at Appendix A lines 726-731 and implemented at Appendix
4 C lines 55-68. Thus, this embodiment prevents users from sending
5 a document to a recipient who is not registered with the system
6 administrator. It also allows users of the system to address
7 messages without typing the recipient's name. According to
8 another embodiment of the invention, shown and described below
9 with reference to Figure 10, users are permitted to send email to
10 any internet address and a list box is optionally used to display
11 an address book. As seen in Figure 3, the reply button 82 in the
12 button bar 80 is grayed out because that function is not available
13 when authoring a text message. This is accomplished in Appendix A
14 at lines 1422-1438.

15
16 Figures 5-8 illustrate the presently implemented rebus
17 component which is described in detail in Appendix D. The rebus
18 component presents a window 104 which includes a "to:" field 106,
19 a message area 108, a "T" button 110 for selecting a template
20 sentence, an "S" button 112 for hiding/displaying symbols, a "-"
21 button 114 for hiding/displaying guesses, and a scrollable area
22 116 to the right of the message area which displays sets of
23 symbols to be used in coding a rebus. In addition, the rebus
24 component displays several pull down menus which are particular to

1 it. These include the template choices menu 118 and the symbol
2 choices menu 120.

3
4 According to the presently implemented embodiment which is
5 detailed in Appendix D, the author of a rebus begins by selecting
6 a template sentence from a selection of sentences which are
7 capable of being expressed as a rebus using the symbol sets
8 provided. The template selection may be made via the T button 110
9 or the pull down menu 118. When a template sentence is selected,
10 a suggested set of symbols is displayed in the field 116.

11 Different symbols may be viewed by selecting a symbol set from the
12 Symbol Choices menu 120. As illustrated in Figure 8, symbols are
13 grouped according to the kinds of words they symbolize such as
14 "action symbols", "object symbols", "quantity symbols", etc. As
15 with other components of the KIDCODE® program, and as shown in
16 Figure 6, the "to:" field 106 presents a drop down or pop up list
17 box through which the email is addressed by selecting a registered
18 user.

19
20 The author of the rebus codes the template sentence by
21 dragging symbols from the scrollable field 116 to the message area
22 108. This is best illustrated by Figure 7. Symbols, when placed
23 on a coded word in the template sentence, will snap into place
24 when they are dragged into the area 108. According to the

1 invention, not every word in the template sentence is designed to
2 be coded with a symbol. According to the presently preferred
3 embodiment, words which are to be coded appear in red text. For
4 example, as shown in Figure 7, two symbols have been dragged into
5 the message area and have snapped over the now hidden words
6 "threw" and "window". The words "ball and "through" are also red
7 text and can be coded with a proper symbol. The author can hide
8 the symbols and display the words in the sentence which are
9 covered by symbols by clicking on the "S" button 112. (Appendix D
10 lines 2433-2451 and 1348-1349.) However, when the recipient
11 receives the rebus, clicking on the button "S" will not reveal the
12 words beneath the symbols, but will only make the symbols
13 disappear. This is an example of how user "roles" alter the tools
14 available in a component. See Appendix D lines 1351-1365. When
15 the author finishes coding the rebus, he or she clicks on the send
16 button 84. The main email component then automatically encodes
17 the rebus as a MIME attachment to Internet mail and sends the mail
18 to the recipient's mailbox. See Appendix F.

19
20 Turning now to Figure 8, when the recipient of the rebus
21 opens the email message containing a rebus, the KIDCODE® main
22 email component automatically decodes the MIME attachment,
23 determines that it is a rebus, and opens it in the rebus reading
24 component. See Appendix F. The message appears with empty text

1 boxes (e.g. 122, 124, 126) beneath the graphic symbols. The
2 recipient of the message must solve the rebus by typing in the
3 text boxes the words which he/she believes are represented by the
4 graphic symbols. As mentioned above, the "-" button 114 is for
5 hiding/displaying the guesses typed in the boxes. When the
6 recipient has typed in words for all the graphic symbols, he/she
7 clicks on the reply button 82 to send the solution back to the
8 author. Figure 8 shows the screen after the button 82 has been
9 clicked. Thus it is grayed out to prevent the same message from
10 being sent twice. The palette 116 is available to the decoder for
11 browsing only. The features which allow symbols to be placed on
12 the message are disabled for the decoder.

13
14 One of the authoring/reading components of the invention is a
15 workbook message handler, an example of which is illustrated in
16 Figure 9. The screen shot shown in Figure 9 illustrates the
17 "student role" of a workbook message handler. The window 130 of
18 the student role workbook message handler preferably includes
19 "to:" and "from:" fields 132, 134 which are filled in by the
20 teacher before the message is sent to the student, as well as six
21 fields 136, 138, 140, 142, 144, 146 which must be filled in by the
22 student before the message is returned to the teacher. As shown
23 in Figure 9, the window 130 also includes a title 148, a date 150
24 and various instructions 152. Those skilled in the art will

1 appreciate that the date 150 may be automatically entered when the
2 message is sent to the student. The fields 136, 138, 140, 142,
3 144, 146, the title 148, and the instructions 152 may be manually
4 entered by the teacher or may be selected as part of a template.
5 In other words, the workbook message handler component may be a
6 complex tool which allows teachers to author an infinite number of
7 "problem messages" to students or it may be a modular set of pre-
8 written problems or templates for problems. The workbook message
9 handler component preferably includes many pre-written problems.
10 Additional pre-written problems will be available through
11 additional modular components. One important feature of the
12 workbook message handler components is that they identify user
13 status and automatically present the proper "role" of either
14 teacher or student.

15
16 The KIDCODE® program described above is designed to be easy
17 to use by very young children. Figures 10-17 illustrate another
18 embodiment of the invention which is designed for a more
19 sophisticated user, an older child, or an adult. The interface is
20 more complicated, but offers the user more features. Referring
21 now to Figure 10, the interface of the second embodiment of the
22 invention includes a menubar 200 which lists some standard menus
23 like FILE, EDIT, SETUP, WINDOW, and HELP. The menubar 200 may
24 also include a clock display 202 which is typically supplied by

1 the operating system and a blinking icon 204 which is typically
2 supplied by TCP/IP connection software such as PPP dialup
3 software, to indicate that the computer is connected to the
4 Internet. The menus MAIL and FOLDER in the menubar 200 are
5 particular to the emailing program and the scroll icon 206 is
6 provided by scripting software, typically part of the operating
7 system. Under the MAIL menu, one will find commands such as NEW
8 for creating new mail, REPLY for replying to a mail message being
9 read, FORWARD, etc. Under the FOLDER menu, one will find the
10 names of user created filing cabinets (folders) where incoming
11 mail can be saved. Under the SETUP menu, one will find commands
12 for setting the necessary information to make a connection with
13 the Internet, for storing the user's name and password, for
14 scheduling automatic sending and receiving of mail, for performing
15 automated tasks in response to mail (e.g. for automatically filing
16 certain mail in certain folders, autoresponding to certain mail,
17 etc.), etc. Under the WINDOW menu, the user will have the option
18 of viewing INBOX, OUTBOX, FILING CABINET(s), CONNECTION STATUS,
19 etc. The HELP menu preferably provides a context sensitive
20 alphabetical list of help items which are hot linked to html
21 files.

Figure 10 illustrates the email program with a new outgoing message window 208 opened. The message window includes a standard text message field 210, a standard subject field 212, standard multiple recipient address fields 214, and a variety of buttons. The arrow buttons 216 allow the user to scroll among messages in the outbox. The outbox button 218 drops down a list of items in the outbox, from which items may be selected. The magnifying glass button 220 is used to search the user's address book. The "+" button 222 adds a recipient to the address field 214. The trash button 224 places the outgoing message in the trash and closes the window 208. The clock button 226 brings up a menu to schedule when the message will be sent. The rotating arrow button 227 causes the address fields 214 to disappear/reappear thereby expanding/contracting the size of the message field 210. The send button 228 sends the message to the outbox (if it is scheduled for delivery at another time or if the computer is not connected to the Internet) and sends the message otherwise. The button 230 labelled "K" causes the computer to connect to the Internet. As shown in Figure 10, this button 230 is grayed out because, as indicated by the blinking telephone pole icon 204, the computer is already connected to the Internet. The "send via" button 232 allows the user to select from several usernames, email accounts, etc.

1 The outgoing message window 208 shown in Figure 10 allows the
2 user to send standard Internet mail by typing a message in the
3 window 210. However, according to the invention, the window 208
4 also includes buttons 234, 236, 238, 240, 242, and 244, each of
5 which is linked to an installed authoring/reading component. As
6 described above, the number and nature of the authoring/reading
7 components is modularly changeable. In the example shown in
8 Figure 10, six authoring/reading components are shown to be
9 installed. In practice, more, fewer, and/or different components
10 may be installed. The components shown and described herein are:
11 a word processor authoring/reading component linked to the button
12 234, a painting/drawing authoring/reading component linked to the
13 button 236, a spreadsheet authoring/reading component linked to
14 the button 238, a database authoring/reading component linked to
15 the button 240, an image editor authoring/reading component linked
16 to the button 242, and a presentation authoring/reading component
17 linked to the button 244.

18
19 Turning now to Figure 11, when the user clicks on the button
20 234, the word processor component is invoked and it causes a new
21 menubar 250 and a ruler 252 to appear inside the message field 210
22 of the window 208. The word processor component allows
23 sophisticated formatting of messages which would be impossible in
24 a normal Internet email program. For example, margins can be set

1 using the ruler 252; fonts can be changed using the FONT menu from
2 the menubar 250; tables can be created and inserted using the
3 TABLE menu from the menubar 250. In general, the menubar 250
4 provides much or all of the functionality of a full featured word
5 processor program. Those skilled in the art will appreciate that
6 the word processor interface shown in Figure 11 is similar to the
7 interface of Microsoft® Word® 98. It will be noted that the
8 menubar 250 provides a separate HELP menu in addition to the HELP
9 menu provided on the menubar 200. It will be appreciated that the
10 HELP menu could be omitted from the menubar 250 and the help files
11 for the word processor component could be accessed from the main
12 HELP menu on the menubar 200. It will also be noted that when the
13 word processor component is invoked, the button 234 is grayed.

14
15 After a user creates a message with the word processor
16 component, the addressing and mailing procedure is the same as
17 sending an ordinary email. There is no need to save a file,
18 encode it, or attach it to an email message. The main email
19 component of the invention seamlessly performs all of the saving,
20 encoding, and attaching without any of this being exposed to the
21 user. More particularly, the authoring component and the main
22 email component cooperate to save the authored document as a file
23 on the user's disk. See Appendix E lines 229-238 and Appendix A
24 lines 1293-1333 and 329-450. The main email component encodes the

1 file in the MIME format with as many parts as necessary, and sends
2 the MIME file(s) as Internet email message(s). See Appendix F.
3 When the message is received by a person using a copy of the email
4 program of the invention, the receiver's main email component
5 seamlessly concatenates the MIME parts, decodes the MIME file
6 (Appendix F), determines that it is a message created with the
7 word processing component (Appendix A lines 690-694), invokes the
8 word processing component (Appendix A lines 1019-1054), and opens
9 the message with the word processing component (Appendix A lines
10 603-614). The receiver of the message does not have to download
11 any file, find any attachment, execute any decoders, or launch any
12 word processor to see the fully formatted document created by the
13 sender.

14
15 Turning now to Figure 12, when the user clicks on the button
16 236, the painting/drawing component is invoked and it causes a new
17 menubar 260 and a tool palette 262 to appear inside the message
18 field 210 of the window 208. The painting/drawing component
19 allows the author to create a painting (bitmap) graphic or a
20 drawing (vectormap) graphic and send it to another user for
21 viewing/editing. Those skilled in the art will appreciate that
22 the menubar 260 and palette 262 shown in Figure 12 contain the
23 menus and tool icons typically found in a full featured
24 drawing/painting program. Those skilled in the art will

1 appreciate that the painting/drawing component interface shown in
2 Figure 12 is similar to the interface of Aldus® SuperPaint® 3.5.
3 It will be noted that the menubar 260 provides a separate HELP
4 menu in addition to the HELP menu provided on the menubar 200. It
5 will be appreciated that the HELP menu could be omitted from the
6 menubar 260 and the help files for the painting/drawing component
7 could be accessed from the main HELP menu on the menubar 200. It
8 will also be noted that when the painting/drawing component is
9 invoked, the button 236 is grayed.

10
11 After a user creates a graphic image with the
12 painting/drawing component, the addressing and mailing procedure
13 is the same as sending an ordinary email. There is no need to
14 save a file, encode it, or attach it to an email message. The
15 main email component of the invention seamlessly performs all of
16 the saving, encoding, and attaching without any of this being
17 exposed to the user. See Appendices A, E and F. When the message
18 is received by a person using a copy of the email program of the
19 invention, the receiver's main email component seamlessly
20 concatenates MIME parts, decodes the MIME file, determines that it
21 is a message created with the painting/drawing component, invokes
22 the painting/drawing component, and opens the message with the
23 painting/drawing component. The receiver of the message does not
24 have to download any file, find any attachment, execute any

1 decoders, or launch any painting/drawing program to view/edit the
2 graphic image created by the sender. See Appendices A, E and F.

3
4 Turning now to Figure 13, when the user clicks on the button
5 238, the spreadsheet component is invoked and it causes a new
6 menubar 270, a grid 272, and a tool palette 274 to appear inside
7 the message field 210 of the window 208. The spreadsheet
8 component allows the author to create a spreadsheet and send it to
9 another user for viewing/editing. Those skilled in the art will
10 appreciate that the menubar 270 and palette 274 shown in Figure 13
11 contain the menus and tool icons typically found in a full
12 featured spreadsheet program. Those skilled in the art will
13 appreciate that the interface of the spreadsheet component shown
14 in Figure 13 is similar to the interface of Microsoft® Excel® 98.
15 It will be noted that the menubar 270 provides a separate HELP
16 menu in addition to the HELP menu provided on the menubar 200. It
17 will be appreciated that the HELP menu could be omitted from the
18 menubar 270 and the help files for the spreadsheet component could
19 be accessed from the main HELP menu on the menubar 200. It will
20 also be noted that when the spreadsheet component is invoked, the
21 button 238 is grayed.

22
23 After a user creates a spreadsheet with the spreadsheet
24 component, the addressing and mailing procedure is the same as

1 sending an ordinary email. There is no need to save a file,
2 encode it, or attach it to an email message. The main email
3 component of the invention seamlessly performs all of the saving,
4 encoding, and attaching without any of this being exposed to the
5 user. See Appendices A, E and F. When the message is received by
6 a person using a copy of the email program of the invention, the
7 receiver's main email component seamlessly concatenates MIME
8 parts, decodes the MIME file, determines that it is a message
9 created with the spreadsheet component, invokes the spreadsheet
10 component, and opens the message with the spreadsheet component.
11 The receiver of the message does not have to download any file,
12 find any attachment, execute any decoders, or launch any
13 spreadsheet program to view/edit the spreadsheet created by the
14 sender. See Appendices A, E and F.

15
16 Turning now to Figure 14, when the user clicks on the button
17 240, the database component is invoked and it causes a new menubar
18 280, a record selection tool 282, and a free form space 284 to
19 appear inside the message field 210 of the window 208. The
20 database component allows the author to create a database and one
21 or more reports and forms associated with the database and send it
22 to another user for viewing/editing. Those skilled in the art
23 will appreciate that the button bar 286 and the data fields 288
24 are defined by the author of the database using authoring tools

1 found in the menus of the menubar 280. In fact, those skilled in
2 the art will appreciate that the database interface shown in
3 Figure 14 is similar to the interface of Filemaker®Pro 3.0. It
4 will be noted that the menubar 280 provides a separate HELP menu
5 in addition to the HELP menu provided on the menubar 200. It will
6 be appreciated that the HELP menu could be omitted from the
7 menubar 280 and the help files for the database component could be
8 accessed from the main HELP menu on the menubar 200. It will also
9 be noted that when the database component is invoked, the button
10 240 is grayed.

11
12 After a user creates a database with the database component,
13 the addressing and mailing procedure is the same as sending an
14 ordinary email. There is no need to save a file, encode it, or
15 attach it to an email message. The main email component of the
16 invention seamlessly performs all of the saving, encoding, and
17 attaching without any of this being exposed to the user. See
18 Appendices A, E, and F. When the message is received by a person
19 using a copy of the email program of the invention, the receiver's
20 main email component seamlessly concatenates MIME parts, decodes
21 the MIME file, determines that it is a message created with the
22 database component, invokes the database component, and opens the
23 message with the database component. The receiver of the message
24 does not have to download any file, find any attachment, execute

1 any decoders, or launch any database program to view/edit the
2 database created by the sender. See Appendices A, E, and F.

3
4 Turning now to Figure 15, when the user clicks on the button
5 242, the image editing component is invoked and it causes a new
6 menubar 290 and a floating tool palette 292 to appear inside the
7 message field 210 of the window 208. The image editing component
8 allows the author to edit an image and send it to another user for
9 viewing and/or further editing. Those skilled in the art will
10 appreciate that the menubar 290 and palette 292 shown in Figure 15
11 contain the menus and tool icons typically found in a full
12 featured image editing program. Those skilled in the art will
13 appreciate that the interface of the image editing component shown
14 in Figure 15 is similar to the interface of Adobe® Photoshop® 3.5.
15 It will be noted that the menubar 290 provides a separate HELP
16 menu in addition to the HELP menu provided on the menubar 200. It
17 will be appreciated that the HELP menu could be omitted from the
18 menubar 290 and the help files for the database component could be
19 accessed from the main HELP menu on the menubar 200. It will also
20 be noted that when the database component is invoked, the button
21 242 is grayed. Those skilled in the art will appreciate that
22 image editing software is typically not used to create an image
23 but to edit an image created by some other hardware/software such
24 as a digital camera or a scanner. As such, there is typically a

1 menu item for opening or capturing an image. As shown in Figure
2 15, open/capture commands may be found under the FILE menu in the
3 menubar 200. Alternatively, image acquisition commands may be
4 found under a menu item in the menubar 290.

5
6 After a user edits an image with the image editor component,
7 the addressing and mailing procedure is the same as sending an
8 ordinary email. There is no need to save a file, encode it, or
9 attach it to an email message. The main email component of the
10 invention seamlessly performs all of the saving, encoding, and
11 attaching without any of this being exposed to the user. See
12 Appendices A, E, and F. When the message is received by a person
13 using a copy of the email program of the invention, the receiver's
14 main email component seamlessly concatenates MIME parts, decodes
15 the MIME file, determines that it is a message created with the
16 image editor component, invokes the image editor component, and
17 opens the message with the image editor component. The receiver
18 of the message does not have to download any file, find any
19 attachment, execute any decoders, or launch any image editor
20 program to view/edit the image edited by the sender. See
21 Appendices A, E, and F.

22
23 Turning now to Figure 16, when the user clicks on the button
24 244, the presentation (slide show) component is invoked and it

1 causes a new menubar 300, a floating wizard palette 302, and a
2 blank template 304 to appear inside the message field 210 of the
3 window 208. The presentation component allows the author to
4 create a slide show presentation and send it to another user for
5 viewing and/or editing. Those skilled in the art will appreciate
6 that the menubar 300, palette 302, and template 304 shown in
7 Figure 16 are typical of those found in a full featured
8 presentation program. In fact, those skilled in the art will
9 appreciate that the interface of the presentation component shown
10 in Figure 16 is similar to the interface of Microsoft® PowerPoint®
11 98. It will be noted that the menubar 300 provides a separate
12 HELP menu in addition to the HELP menu provided on the menubar
13 200. It will be appreciated that the HELP menu could be omitted
14 from the menubar 290 and the help files for the database component
15 could be accessed from the main HELP menu on the menubar 200. It
16 will also be noted that when the database component is invoked,
17 the button 244 is grayed.

18
19 After a user creates a presentation with the presentation
20 component, the addressing and mailing procedure is the same as
21 sending an ordinary email. There is no need to save a file,
22 encode it, or attach it to an email message. The main email
23 component of the invention seamlessly performs all of the saving,
24 encoding, and attaching without any of this being exposed to the

1 user. See Appendices A, E, and F. When the message is received
2 by a person using a copy of the email program of the invention,
3 the receiver's main email component seamlessly concatenates MIME
4 parts, decodes the MIME file, determines that it is a message
5 created with the presentation component, invokes the presentation
6 component, and opens the message with the presentation component.
7 The receiver of the message does not have to download any file,
8 find any attachment, execute any decoders, or launch any
9 presentation program to view/edit the presentation created by the
10 sender. See Appendices A, E, and F.

11
12 As described above, messages received by the email software
13 according to the invention are seamlessly decoded and displayed.
14 Figure 17 illustrates an incoming message window 408 which
15 displays a message containing a combination of text and graphics
16 in the message field 410. The incoming message window 408 also
17 includes a subject field 412 and a "from:" address field 414 which
18 includes information about the time the message was sent and
19 received. Arrow buttons 416 allow the user to scroll through
20 messages in the "in box". Button 418 drops a menu list of
21 messages in the in box from which a message may be selected. The
22 "+" button 420 adds the sender's address to the recipient's
23 address book. The rotating arrow 427 hides the address field 414

1 and expands the message field 410. Buttons 428 and 430 are not
2 implemented, but may be used for public key decryption, etc.

3
4 As mentioned above, the modular components of the invention
5 may be authoring/reading components or read only components.
6 Figure 17 illustrates an incoming message window 408 which
7 displays a message containing a combination of text and graphics
8 in message field 410 without any editing/authoring tools. The
9 message may have been created with the word processing component
10 or the painting and drawing component. The component used to
11 create the message need not be known by the recipient of the
12 message when it is opened with a read only component as shown. It
13 will be appreciated that the message could also be automatically
14 opened with an authoring/reading component, in which case, the
15 message field 410 in Figure 17 would also include a menubar, and
16 perhaps a tool palette. According to the invention, the email
17 client software may be provided with a full complement of read
18 only components and the authoring components may be installed
19 according to the user's choices. Additionally, the email client
20 software may be programmed to automatically download a reading
21 component from an ftp site when it encounters a message which
22 requires a component which is not yet installed.

23

1 There have been described and illustrated herein several
2 embodiments of electronic mail software with modular integrated
3 authoring/reading software components. While particular
4 embodiments of the invention have been described, it is not
5 intended that the invention be limited thereto, as it is intended
6 that the invention be as broad in scope as the art will allow and
7 that the specification be read likewise. Thus, while particular
8 graphical interfaces have been disclosed, it will be appreciated
9 that other interfaces could be utilized. Also, while particular
10 authoring/reading components have been shown, it will be
11 recognized that other types of authoring/reading components could
12 be provided in the spirit of the invention. Moreover, while
13 particular configurations have been disclosed in reference to the
14 code in the appendices, it will be appreciated that other
15 configurations could be used as well. Further, while particular
16 software code and pseudocode have been disclosed to perform
17 various functions, it will be appreciated that other code and/or
18 hardware could be utilized to accomplish those functions and
19 should be considered the equivalents thereof. It will therefore
20 be appreciated by those skilled in the art that yet other
21 modifications could be made to the provided invention without
22 deviating from its spirit and scope as so claimed.